

REMARKS

The Examiner is thanked for the performance of a thorough search.

No claims have been canceled, added, or amended. Hence, Claims 1-23, 47, and 49-72 are pending in the present application.

The issues raised in the final Office Action mailed May 25, 2010 are addressed hereinafter.

I. ISSUES RELATED TO THE CITED ART

A. INDEPENDENT CLAIM 1

Claim 1 was rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over Abrams et al., U.S. Patent No. 6,675,350 (“ABRAMS”) in view of Schaeck, U.S. Patent No. 7,502,833 (“SCHAECK”). The rejection is respectfully traversed.

(1). Claim 1 includes the feature of a page parameter associated with a web page.

It is respectfully submitted that without having a clear understanding of what a page parameter is, it is not possible to accurately examine Claim 1.

As described in the present application, a page parameter is a named variable associated with a web page, where such named variable can be assigned different values during different invocations to display the web page. (See, for example, Figs. 1-2 and 3A, and paragraphs [0024]-[0025], [0029], [0032]-[0033], [0039], [0052] of the present application.) This difference between a page parameter and a value of the page parameter is expressly indicated in the feature of Claim 1 of “passing a value associated with the page parameter to the portlet as a value of the portlet parameter”. This feature of Claim 1 also makes it clear that the portlet parameter is an entity that is different than the page parameter because, while the page parameter is associated with a web page, the portlet parameter is associated with a portlet. It is noted that typically the name that a web page uses for a parameter is different from the name that a portlet uses for its

parameter(s) because web pages are typically designed by different people than the portlet developers who develop portlets that are provided through some kind of a web service. (See paragraphs [0003], [0006], and [0013] of the present application; for an example, see Fig. 1 of the present application, in which “PORTA_P1” is the name portlet “A” uses for one of its parameters, while “PAGEA_P1” is the name web page “A” uses for a corresponding page parameter.)

With the above in mind, it is respectfully submitted that that several express features of Claim 1 are not described by ABRAMS and SCHAECK. These are addressed in separate sections below.

- (2). SCHAECK does not describe or suggest the feature of Claim 1 of generating and storing a mapping that maps one or more page parameters to one or more portlet parameters, where the mapping is stored separate from web pages associated with the one or more page parameters.

Among other features, Claim 1 comprises the feature of:

generating and storing a mapping that maps one or more page parameters to one or more portlet parameters, wherein the mapping is stored separate from web pages associated with the one or more page parameters.

The Office Action asserts that this feature of Claim 1 is described in SCHAECK in col. 1, lines 59-61 and in col. 4, lines 17-23 and 45-47. This assertion is incorrect.

SCHAECK describes a method for dynamically integrating remote portlets into a local portal server by providing the portal server with a portal registry component and a portlet proxy component, where the portal registry component provides a service description of a remote portlet to the portlet proxy component which establishes a SOAP-communication with the remote portal server that provides the remote portlet. (See SCHAECK, Fig. 5 and col. 3, lines 19-37.) Specifically, with respect to its Fig. 5, in col. 4, lines 11-26 SCHAECK describes that when a portal server imports a portlet description of a remote portlet, remote portlet access information (ref. 20 in Fig. 5) is created; when a page that has the remote portlet is displayed, the

portal server creates and invokes a portal proxy configured with the remote portlet access information to call the remote portlet on the remote portal server. Thus, Fig. 5 and this passage from SCHAECK suggest that the remote portlet access information includes the description of the remote portlet but does not include any information that is specific to any particular page (much less the names that specific pages associate with specific parameters). This is because SCHAECK makes it very clear that a remote portlet is registered locally only once, but can be invoked for many different pages by creating remote proxy processes that are configured with the remote access information for the registered remote portlet. (See also SCHAECK, Fig. 7H and col. 5, lines 45-55, which describe that once registered with a portal server, the remote portlet is made available to users of the portal server.)

In contrast, Claim 1 expressly features generating and storing a mapping that maps one or more page parameters to one or more portlet parameters, where the mapping is stored separate from web pages associated with the one or more page parameters. Since, as discussed above, a page parameter is a named variable associated with a specific web page, and since the remote portlet access information in SCHAECK does not include any information that is specific to any particular web page, SCHAECK does not describe this feature of Claim 1.

On page 5, the final Office Action asserts that in col. 1, lines 59-61 SCHAECK describes a web page having page parameters. This assertion is incorrect. In col. 1, lines 58-64, SCHAECK states:

The HR portlet uses a HR Web-Service to calculate variable pay. A possible implementation is a portlet that by default displays a form to query the required parameters, e.g., the position and last grade of an employee. When the employee enters his data at the HR portlet, it invokes the remote Web-Service to calculate the variable pay based on that data.

The above passage of SCHAECK makes it very clear that any data entered by a user is actually associated with portlet parameters, and that the user enters the data into the portlet through a

query form provided by the portlet. Thus, this passage of SCHAECK describes that a portlet receives values directly through a query form of the portlet and not through any page parameters that are associated with a web page. To the extent that a portlet may be included in a web page, in col. 4, lines 43-47 and in col. 5, lines 59-67 SCHAECK describes that a user can select portlets registered in a portal server and can add references to the selected portlets in the user's personal page. Thus, these passages of SCHAECK suggest that a portlet is "hardcoded" in a user's personal web page in the sense that, when the user's web page is displayed, the portal server can use the references within the web page to invoke the portlet (either directly for local portlets or through a portlet proxy as described in col. 4, lines 17-23) without the need to use any separate mapping that maps page parameters to portlet parameters (as featured in Claim 1).

For the foregoing reasons, SCHAECK does not describe or suggest the feature of Claim 1 of generating a mapping that maps one or more page parameters to one or more portlet parameters, where the mapping is stored separate from pages associated with the one or more page parameters.

Finally, it is noted that the final Office Action includes an inconsistency related to the above feature of Claim 1. For example, in page 4 the final Office Action asserts that ABRAMS does not describe a mapping of page parameters to portlet parameters. However, in response to the Applicants' previous arguments, in pp. 27-28 the final Office Action asserts that in col. 6, lines 12-25 ABRAMS describes such mapping. Specifically, in pp. 27-28 the final Office Action asserts that the HTML code that is generated for formatting and displaying the stored summary information/headlines in a user-customized "MyPortal" view reads on a mapping such as the mapping in Claim 1. This last assertion is incorrect. The above feature of Claim 1 expressly recites generating a mapping that maps one or more page parameters to one or more portlet parameters. In contrast, the above passage of ABRAMS describes the generation of customized

HTML code for formatting and displaying information that has already been extracted from web pages. Since customly-generated HTML code has nothing to with page parameters, and since ABRAMS does not even describe portlets or any functionality of using portlets, it is respectfully submitted that ABRAMS in general and the above passage from ABRAMS in particular do not describe or suggest the feature of Claim 1 of generating a mapping that maps one or more page parameters to one or more portlet parameters, where the mapping is stored separate from pages associated with the one or more page parameters.

- (3). ABRAMS and SCHAECK do not describe or suggest the feature of Claim 1 of determining that the web page is associated with a page parameter from the one or more page parameters.

Among other features, Claim 1 comprises the feature of:

determining that the web page is associated with a page parameter from the one or more page parameters.

The final Office Action asserts that this feature of Claim 1 is described by a combination of ABRAMS and SCHAECK. Specifically, the final Office Action asserts that this feature of Claim 1 is described in col. 4, lines 13-30 of ABRAMS and in col. 1, lines 59-61 and in col. 5, lines 45-50 of SCHAECK. This assertion is incorrect.

In col. 4, lines 13-30 ABRAMS describes the operation of an HTML parser tool (i.e., a standalone program), where a user can enter a URL address of a web site into the HTML parser tool. The HTML parser tool displays the selected web site in a GUI pane, and allows the user to select a text from that GUI pane and to specify that the selected text is to be used as a constraint filter to reduce the number of headline links displayed in another GUI pane. Thus, since the user can enter any arbitrary URL address in the HTML parser tool, the field in which the user enters the URL address is associated with a parameter of the parser tool and not with any parameter associated with a specific web page. Further, to the extent this passage of ABRAMS describes

any functionality, such functionality relates to receiving user input as a program parameter in a standalone software program such as ABRAMS' HTML parser tool.

As discussed above, in col. 1, lines 58-64, SCHAECK describes that a portlet receives values directly through a query form of the portlet and not through any page parameters that are associated with a web page. Further, in col. 5, lines 45-50 SCHAECK describes that a portal server displays service details of a remote portlet and allows an administrator to use an "Add" button to add the remote portlet to that portal server's portlet registry. Significantly, however, as discussed above, the remote portlet access information stored in the portal server's registry does not include any information that is specific to any particular page. To the extent that these passages of SCHAECK describe any functionality, such functionality relates to registering remote portlet access information for a particular portlet and has nothing to do with page parameters associated with a specific web page.

Thus, the combination of the above passages of ABRAMS and SCHAECK does not describe or suggest anything relating to page parameters associated with a web page or to a mapping that maps such page parameters to portlet parameters. Further, it is not clear how the functionalities described in the above passages of ABRAMS and SCHAECK can be combined because receiving user input as a program parameter of a standalone application has nothing to do with registering remote portlets in a portal server. Moreover, even if the functionalities of ABRAMS and SCHAECK described above are somehow combined, the combination would not describe or even remotely suggest any functionality or use of page parameters associated with a specific web page.

In contrast, the above feature of Claim 1 indicates a functionality of determining that the web page is associated with a page parameter from the one or more page parameters that are specified in a mapping that maps the page parameters to portlet parameters. Since as discussed

above the combination of ABRAMS and SCHAECK does not describe any functionality or use of page parameters associated with a web page, ABRAMS and SCHAECK do not describe or suggest the above feature of Claim 1. Further, as discussed above, in col. 4, lines 43-47 and in col. 5, lines 59-67 SCHAECK suggests that a portlet is “hardcoded” in a user’s personal web page in the sense that, when the user’s web page is displayed, the portal server can use the references within the web page to invoke the portlet (either directly for local portlets or through a portlet proxy as described in col. 4, lines 17-23) without the need to use any separate mapping that maps any page parameters to portlet parameters (as featured in Claim 1).

For the foregoing reasons, the combination of ABRAMS and SCHAECK does not describe or suggest the feature of Claim 1 of determining that the web page is associated with a page parameter from the one or more page parameters.

- (4). ABRAMS and SCHAECK do not describe or suggest the feature of Claim 1 of wherein using the mapping includes retrieving and inspecting the mapping to determine that the page parameter is mapped to a portlet parameter of a portlet.

Among other features, Claim 1 comprises the features of:

wherein using the mapping includes retrieving and inspecting the mapping to determine that the page parameter is mapped to a portlet parameter of a portlet.

The final Office Action asserts that this feature of Claim 1 is described by a combination of ABRAMS and SCHAECK. Specifically, the final Office Action asserts that this feature of Claim 1 is described in col. 4, lines 13-30 of ABRAMS and in col. 4, lines 16-26 and 45-55 of SCHAECK. This assertion is incorrect.

As discussed above, in col. 4, lines 13-30 ABRAMS describes that a user can enter a URL address in the HTML parser tool, where the field in which the user enters the URL address is associated with a parameter of the parser tool and not with any parameter associated with a specific web page. Further, to the extent this passage of ABRAMS describes any functionality,

such functionality relates to receiving user input as a program parameter in a standalone software program such as ABRAMS' HTML parser tool.

As also discussed above, in col. 4, lines 11-26 and 45-50 SCHAECK describes that when a portal server imports a portlet description of a remote portlet, remote portlet access information (ref. 20 in Fig. 5) is created; however, this remote portlet access information does not include any information that is specific to any particular page because SCHAECK makes it clear that a remote portlet is registered locally only once, but can be invoked for many different pages by creating remote proxy processes that are configured with the remote access information for the registered remote portlet. Further, to the extent these passages from SCHAECK describe any functionality, such functionality relates to displaying a web page that references a remote portlet by invoking the portlet based on remote portlet access information that does not include any mapping of portlet parameters to the parameters of that specific web page. This is because, as discussed above, at least in col. 4, lines 43-47 and in col. 5, lines 59-67 SCHAECK describes that a user can select portlets registered in a portal server and can add references to the selected portlets in the user's personal page, which suggests that a portlet is "hardcoded" in a user's personal web page in the sense that, when the user's web page is displayed, the portal server can use the references in the web page to invoke the portlet (either directly for local portlets or through a portlet proxy) without the need to use any mapping that maps page parameters to portlet parameters.

Thus, the combination of the above passages of ABRAMS and SCHAECK does not describe or suggest anything related to inspecting or otherwise using a mapping to determine which portlets and portlet parameters thereof are mapped to the page parameter associated with a specific web page. Further, it is not clear how the functionalities described in the above passages of ABRAMS and SCHAECK can be combined because receiving user input as a program

parameter of a standalone application has nothing to do with using remote portlet access information to invoke remote portlets when pages that reference the remote portlets are displayed. Moreover, even if the functionalities of ABRAMS and SCHAECK described above are somehow combined, the combination would not describe or even remotely suggest any functionality related to inspecting or otherwise using a mapping to determine which portlets and portlet parameters thereof are mapped to the page parameter associated with a specific page. To put it differently, the functionalities described in the above passages of ABRAMS and SCHAECK do not suggest to one of skill in the art any need or any other reason for making use of a mapping that maps page parameters to portlet parameters; this is at least because SCHAECK describes a comprehensive mechanism through which a portal server can use the references within a web page to invoke a remote portlet without the need to use any separate mapping that maps page parameters to portlet parameters.

In contrast, the above feature of Claim 1 indicates a functionality of inspecting a mapping that maps page parameters to portlet parameters to determine that the page parameter of a web page is mapped to a portlet parameter of a particular portlet. Since as discussed above the combination of ABRAMS and SCHAECK does not describe any functionality or use of a mapping that maps page parameters to portlet parameters, ABRAMS and SCHAECK do not describe or suggest the above feature of Claim 1.

For the foregoing reasons, the combination of ABRAMS and SCHAECK does not describe or suggest the feature of Claim 1 of wherein using the mapping includes retrieving and inspecting the mapping to determine that the page parameter is mapped to a portlet parameter of a portlet.

- (5). ABRAMS and SCHAECK do not describe or suggest the feature of Claim 1 of passing a value associated with the page parameter to the portlet as a value of the portlet parameter.

Among other features, Claim 1 comprises the features of:

passing a value associated with the page parameter to the portlet as a value of the portlet parameter.

The final Office Action asserts that this feature of Claim 1 is described by a combination of ABRAMS and SCHAECK. Specifically, the final Office Action asserts that this feature of Claim 1 is described in Figs. 2A, 2B, and 6 of ABRAMS and in col. 1, lines 61-64 of SCHAECK. This assertion is incorrect.

In Figs. 2A and 2B, ABRAMS shows a computer screen image of a HTML parser tool showing component window panes and controls as seen by a user. (See ABRAMS, col. 2, lines 36-38 and col. 3, lines 65-66.) In Fig. 6, ABRAMS shows a tabbed display of data summarized and organized by using ABRAMS' invention (i.e., the HTML parser tool) with a list of summaries from a single tab. (See also ABRAMS, col. 5, line 55 to col. 6, line 5.) Thus, these figures and passages of ABRAMS show and describe a standalone software application (i.e., a computer program composed of executable code) that has nothing to do with portlets. To the extent that on the bottom of page 3 and on the top of page 4 the final Office Action asserts that Figs. 2A and 2B show that “the portlets use page parameters such as URL data to display page/summary information for the page located at the particular URL...”, such assertion is incorrect because it is **not factually** supported by anything described or suggested in ABRAMS. For example, as discussed above, ABRAMS describes a standalone computer program that has nothing to do with using portlets, and the URL entered by a user in ABRAMS' standalone program cannot possibly correspond to a page **parameter** associated with a web page.

As also discussed above, in col. 1, lines 58-64, SCHAECK describes that a portlet receives values directly through a query form of the portlet and not through any page parameters that are associated with a web page. Thus, this passage of SCHAECK does not describe or suggest any functionality that involves passing the value of a page parameter of a web page as

the value of a portlet parameter of a portlet. This is not the least because the above passage of SCHAECK describes that the user enters the values of the **portlet** parameters directly into a query form provided by the **portlet itself**. It is noted that in page 6, lines 3-5, the final Office Action asserts that “certain data for a page is passed to a portlet parameter, through mapping data, such that the appropriate portlet proxy and SOAP calls are made to the portlet.” This assertion is incorrect because, as discussed above, SCHAECK does not describe any mapping that maps page parameters to portlet parameters. To the extent that in the above assertion the final Office Action refers to the remote portlet access information (col. 4, lines 13 and 19-20) as “mapping data”, such assertion is incorrect because as discussed above the remote portlet access information of SCHAECK does not include any information about page parameters of specific web pages. Further, SCHAECK expressly describes that the SOAP protocol deals with how a program in one computer can call a program in another computer (see SCHAECK, Figs. 3-4 and col. 2, lines 12-25), and therefore has nothing to do with mapping page parameters to portlet parameters.

Thus, it is noted that the combination of the above passages of ABRAMS and SCHAECK does not describe or suggest anything related passing a value associated with a page parameter to a portlet. Specifically, it is noted that as a standalone program the HTML parser tool of ABRAMS has nothing to do with invoking portlets, and the functionality of invoking a remote portlet in SCHAECK does not involve passing the value of a page parameter to a portlet because SCHAECK describes a comprehensive mechanism through which a portal server can use the references within a web page to invoke a remote portlet **without** the need to use any separate mapping that maps page parameters to portlet parameters. Thus, even if the functionalities described in the above passages of ABRAMS and SCHAECK are somehow combined, the combination would not describe or even remotely suggest passing the value of a

page parameter to a portlet.

In contrast, the above feature of Claim 1 indicates a functionality of passing a value associated with the page parameter to the portlet as a value of the portlet parameter. This feature of Claim 1 clearly indicates that the value associated with a page parameter is different than the page parameter itself. Further, when considered along with the other features of Claim 1, this feature of Claim 1 clearly indicates a functionality in which, during the process of displaying a web page, the portlet and the portlet parameter thereof (to which the value of the page parameter is passed) are determined from a mapping that maps page parameters to portlet parameters. Since as discussed above the combination of ABRAMS and SCHAECK does not describe any functionality of passing the value of a page parameter to a portlet, ABRAMS and SCHAECK do not describe or suggest the above feature of Claim 1.

For the foregoing reasons, the combination of ABRAMS and SCHAECK does not describe or suggest the feature of Claim 1 of passing a value associated with the page parameter to the portlet as a value of the portlet parameter.

For the reasons provided above in sections (1) through (5), ABRAMS and SCHAECK whether taken alone or in combination do not describe or suggest all features of Claim 1. Thus, Claim 1 is patentable under 35 U.S.C. § 103(a) over ABRAMS in view of SCHAECK. Reconsideration and withdrawal of the rejection of Claim 1 is respectfully requested.

B. INDEPENDENT CLAIM 18

Claim 18 was rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over ABRAMS in view of SCHAECK. The rejection is respectfully traversed.

Among other features, Claim 18 comprises:

generating and storing a first mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters,

wherein the first mapping is stored separate from web pages associated with the one or more page parameters;
...;
in response to a user manipulating a component of the web page, a portlet that previously generated the component generating a particular event;
...;
logic associated with the web page intercepting data, passed by the portlet, that represents the particular event;
retrieving and inspecting the first mapping, wherein inspecting the first mapping includes:
determining, based on the first mapping and the intercepted data, an action to perform in response to the particular event;
determining, based on the first mapping, that an event output parameter associated with the particular event is mapped to a page parameter; and
causing the action to be performed, wherein causing the action to be performed comprises passing a value of the event output parameter as the value of the page parameter;
....

The above features of Claim 18 indicate a first mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters. The above features of Claim 18 also indicate that in response to a user manipulating a component of a web page, a portlet that previously generated the component generates a particular event. Logic associated with the web page intercepts data that represents the particular event, and a first mapping is inspected based on the intercepted data to determine an action to perform in response to the particular event and that an event output parameter of the particular event is mapped to a page parameter. Thus, these features of Claim 18 indicate that the first mapping maps at least one event generated by a portlet to an action and at least one event output parameter to a page parameter. It is respectfully submitted that ABRAMS and SCHAECK do not describe these features of Claim 18.

- (1). SCHAECK does not describe or suggest the feature of Claim 18 of generating and storing a first mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters, where the first mapping is stored separate from web pages associated with the one or more page parameters.

Among other features, Claim 18 comprises the feature of:

generating and storing a first mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters, wherein the first mapping is stored separate from web pages associated with the one or more page parameters.

The Office Action asserts that this feature of Claim 18 is described in SCHAECK in col. 1, lines 59-63 and in col. 4, lines 17-23 and 45-47. This assertion is incorrect.

As discussed above with respect to Claim 1, SCHAECK describes a method for dynamically integrating remote portlets into a local portal server by providing the portal server with a portal registry component and a portlet proxy component, where the portal registry component provides a service description of a remote portlet to the portlet proxy component which establishes a SOAP-communication with the remote portal server that provides the remote portlet. (See SCHAECK, Fig. 5 and col. 3, lines 19-37.) Specifically, with respect to its Fig. 5, in col. 4, lines 11-26 SCHAECK describes that when a portal server imports a portlet description of a remote portlet, remote portlet access information (ref. 20 in Fig. 5) is created; when a page that has the remote portlet is displayed, the portal server creates and invokes a portal proxy configured with the remote portlet access information to call the remote portlet on the remote portal server. Thus, Fig. 5 and this passage from SCHAECK suggest that the remote portlet access information includes the description of the remote portlet but does not include any information that is specific to any particular page. This is because SCHAECK makes it very clear that a remote portlet is registered locally only once, but can be invoked for many different pages by creating remote proxy processes that are configured with the remote access information for the registered remote portlet. (See also SCHAECK, Fig. 7H and col. 5, lines 45-55, which describe that once registered with a portal server, the remote portlet is made available to users of the portal server.)

Further, Fig. 5 and the above passages of SCHAECK do not describe or in any way suggest that the remote portlet access information includes anything about actions, events, and

event output parameters. In fact, SCHAECK does not even mention the terms “action”, “event”, and “event output parameter”, and upon reviewing SCHAECK in its entirety the Applicants could not find anything that corresponds to these features of Claim 18. For example, the final Office Action asserts that in col. 1, lines 62-63 SCHAECK describes “parameters that can be input events causing a event output action.” This assertion completely confuses and obfuscates the difference between “parameters” and “events” because a parameter is something very different than an event. Further, this assertion is factually incorrect because in col. 1, lines 62-63 SCHAECK describes that when an employee enters his data at the HR portlet, the portlet invokes the remote Web-Service to calculate the variable pay based on that data; thus, there is absolutely nothing in this passage that describes or suggests that a portlet generates any events or causes an event output action in response to receiving input data in a query form provided by the portlet. Rather, this passage very clearly describes that the portlet (i.e., the HR portlet) simply uses the input data to perform the processing logic that is encoded in the executable code of the portlet.

In contrast, Claim 18 expressly features generating and storing a first mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters, where the first mapping is stored separate from web pages associated with the one or more page parameters. Thus, the mapping of Claim 18 is a two-fold mapping: (1) events are mapped to actions, and (2) event output parameters are mapped to page parameters. Further, the other features of Claim 18 indicate that at least one event mapped in the mapping is generated by a portlet in response to a user manipulating a web page component that was previously generated by the portlet. Since (as discussed above with respect to Claim 1) a page parameter is a named variable associated with a specific web page, and since the remote portlet access information in SCHAECK does not include any information that is specific to any

particular web page and does not include any information about portlet-generated events and any parameters thereof, SCHAECK does not describe this feature of Claim 18.

It is noted that to the extent that a portlet may be included in a web page, in col. 4, lines 43-47 and in col. 5, lines 59-67 SCHAECK describes that a user can select portlets registered in a portal server and can add references to the selected portlets in the user's personal page. Thus, these passages of SCHAECK suggest that a portlet is "hardcoded" in a user's personal web page in the sense that, when the user's web page is displayed, the portal server can use the references within the web page to invoke the portlet without the need to use any separate mapping that maps events to actions and event output parameters to page parameters (as featured in Claim 18). Further, in col. 4, lines 20-25 SCHAECK expressly describes that to invoke a remote portlet, a portlet proxy acts as a SOAP-client to convert normal method calls into SOAP-remote procedure calls, and that at the remote portal server a SOAP-server converts the received SOAP-remote procedure calls into local method calls to the local portlet. Thus, this mechanism of invoking a remote portlet does not use any event-based functionality, and therefore there is absolutely no need or reason to include any event information in the remote portlet access information that is used to configure the portlet proxy that invokes the remote portlet.

For the foregoing reasons, SCHAECK does not describe or suggest the feature of Claim 18 of generating and storing a first mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters, wherein the first mapping is stored separate from web pages associated with the one or more page parameters.

- (2). ABRAMS does not describe or suggest the feature of Claim 18 of in response to a user manipulating a component of the web page, a portlet that previously generated the component generating a particular event.

Among other features, Claim 18 comprises the feature of:

in response to a user manipulating a component of the web page, a portlet that previously generated the component generating a particular event.

The final Office Action asserts that this feature of Claim 18 is described in col. 4, lines 20-21 of ABRAMS. Specifically, the final Office Action asserts that “whereas a user manipulates a web address in a component 220 of a portlet, causing the portlet in the page to generate a URL selection event”. This assertion is incorrect.

As discussed above with respect to Claim 1, in col. 4, lines 13-30 ABRAMS describes the operation of an HTML parser tool (i.e., a standalone program), where a user can enter a URL address of a web site into the HTML parser tool. (See also ABRAMS, col. 3, lines 65-66.) The HTML parser tool displays the selected web site in a GUI pane, and allows the user to select a text from that GUI pane and to specify that the selected text is to be used as a constraint filter to reduce the number of headline links displayed in another GUI pane. Significantly, however, a standalone computer program (such as ABRAMS’ HTML parser tool) has nothing to do with using portlets, and the component 220 in Fig. 2A of ABRAMS is a GUI pane configured to receiving a URL entered by a user. (See also ABRAMS, col. 4, lines 7-9.) Since a GUI pane generated by a standalone computer program is not a portlet, and since all the processing logic of standalone computer program is encoded in the executable code of the program, the HTML parser tool of ABRAMS and any components thereof cannot possibly correspond to a portlet that generates events. To assert otherwise is simply a disregard of common sense.

In contrast, the above feature of Claim 18 indicates a portlet generating a particular event in response to a user manipulating a component of the web page, where the component was previously generated by the portlet. Since as discussed above the HTML parser tool of ABRAMS and any components thereof cannot possibly correspond to a portlet that generates events, ABRAMS does not describe the above feature of Claim 1.

For the foregoing reasons, ABRAMS does not describe or suggest the feature of Claim 18 of in response to a user manipulating a component of the web page, a portlet that previously

generated the component generating a particular event.

- (3). ABRAMS and SCHAECK do not describe or suggest the feature of Claim 18 of logic associated with the web page intercepting data, passed by the portlet, that represents the particular event.

Among other features, Claim 18 comprises the features of:

logic associated with the web page intercepting data, passed by the portlet, that represents the particular event.

The final Office Action asserts that this feature of Claim 1 is described by a combination of ABRAMS and SCHAECK. Specifically, the final Office Action asserts that this feature of Claim 1 is described in col. 4, lines 1-12 and col. 6, lines 25-32 of ABRAMS and in col. 1, lines 64-65 of SCHAECK. This assertion is incorrect.

As discussed above, in col. 3, line 65 to col. 4, lines 12 ABRAMS describes the operation of an HTML parser tool (i.e., a standalone program), where a user can enter a URL address of a web site into a GUI pane generated by the HTML parser tool. Significantly, however, a GUI pane generated by a standalone computer program is not a portlet, and since all the processing logic of a standalone computer program is encoded in the executable code of the program, the HTML parser tool of ABRAMS and any components thereof cannot possibly correspond to portlets or to any components that need to intercept event data generated by a portlet. In col. 6, lines 25-32 ABRAMS describes an implementation of a “MyPortal” technique for viewing of summary/headline data that was extracted with ABRAM’s HTML parser tool, where the technique involves simply listing the extracted headlines with their associated links in an HTML file that can be further manipulated using an HTML editor. Thus, this passage of ABRAMS also fails to describe anything that is even remotely related to portlets or to any components that need to intercept event data generated by a portlet. Further, to the extent these passages of ABRAMS describe any functionality, such functionality relates to processing done in a standalone software program (such as ABRAMS’ HTML parser tool) and to storing and manipulating data in an

HTML file.

As also discussed above, in col. 1, lines 58-65, SCHAECK describes that a portlet receives values directly through a query form generated by the portlet, and that the portlet itself displays the result data as a page fragment. Thus, this description expressly indicates that a portlet itself displays the results the portlet generates, which suggests that there is no need for any logic to intercept data generated by the portlet. To the extent that this passage of SCHAECK describes any functionality, such functionality relates to a portlet that directly displays its results as web page fragments.

Thus, the combination of the above passages of ABRAMS and SCHAECK does not describe or suggest anything related to intercepting data generated by a portlet. Further, even if the functionalities of ABRAMS and SCHAECK described above are somehow combined, the combination would not describe or even remotely suggest any functionality related to intercepting data generated by a portlet because ABRAMS describes processing done in a standalone application (which has no need or use of intercepting any portlet-generated data) and SCHAECK describes a portlet displaying its own results (which suggests that there is no need to intercept such results).

In contrast, the above feature of Claim 18 indicates logic associated with the web page intercepting data, passed by the portlet, that represents the particular event. Since as discussed above the combination of ABRAMS and SCHAECK does not describe any functionality or use of related to intercepting data generated by a portlet, ABRAMS and SCHAECK do not describe or suggest the above feature of Claim 18.

For the foregoing reasons, the combination of ABRAMS and SCHAECK does not describe or suggest the feature of Claim 18 of logic associated with the web page intercepting data, passed by the portlet, that represents the particular event.

- (4). ABRAMS does not describe or suggest the feature of Claim 18 of determining, based on the first mapping and the intercepted data, an action to perform in response to the particular event.

Among other features, Claim 18 comprises the feature of:

...wherein inspecting the first mapping includes:
determining, based on the first mapping and the intercepted data, an action to perform in response to the particular event;

The final Office Action asserts that this feature of Claim 18 is described in col. 4, lines 21-24 of ABRAMS. Specifically, the Office Action asserts that the action performed in ABRAMS to display in pane 260 all hyperlinks with their associated text for the selected web site corresponds to the above feature of Claim 18. This assertion is incorrect.

On page 11 the final Office Action expressly concedes that ABRAMS does not describe a mapping that maps events to actions and event output parameters to page parameters, such as the first mapping in Claim 18. Further, as discussed above SCHAECK does not describe generating and storing a mapping such as the first mapping of Claim 18 either. Thus, ABRAMS and SCHAECK whether taken alone or in combination cannot possibly describe any functionality that is based on a mapping such as the first mapping of Claim 18.

Further, in col. 4, lines 21-24 ABRAMS describes that when a user selects a web address in pane 220 of the GUI, the HTML parser tool displays in pane 260 all hyperlinks of the site indicated by the web address. This, however, does not describe or even suggest that any determination to display the hyperlinks is made based on a mapping that maps portlet-generated events to actions and event output parameters to page parameters. In other words, the action of displaying the selected hyperlinks in a window on a computer screen is what the HTML parser tool is programmed to do in its executable code, and therefore such action of displaying has no need or use of any determinations that are made based on a mapping that is equivalent to the first mapping of Claim 18.

In contrast, Claim 18 comprises the feature of determining, based on the first mapping and the intercepted data, an action to perform in response to the particular event. When considered along with the other features of Claim 18, this feature indicates that the determination of an action is made based on a mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters, and on intercepted data that was generated by a portlet in response to a user manipulating a web page component that was previously generated by the portlet. Since ABRAMS does not describe a mapping such as the first mapping of Claim 18 and since the ABRAMS' HTML parser tool does not even need any functionalities that is based on such mapping, ABRAMS does not describe or suggest the above feature of Claim 18.

For the foregoing reasons, it is respectfully submitted that ABRAMS does not describe or suggest the feature of Claim 18 of determining, based on the first mapping and the intercepted data, an action to perform in response to the particular event.

- (5). ABRAMS does not describe or suggest the feature of Claim 18 of determining, based on the first mapping, that an event output parameter associated with the particular event is mapped to a page parameter.

Among other features, Claim 18 comprises:

...wherein inspecting the first mapping includes: ...
determining, based on the first mapping, that an event output parameter associated with the particular event is mapped to a page parameter.

The final Office Action asserts that this feature of Claim 18 is described in col. 4, lines 21-29 of ABRAMS. This assertion is incorrect.

As discussed above, in col. 4, lines 21-30 and with respect to its Figs. 2A and 2B, ABRAMS describes that when a user selects a web address in pane 220 of the GUI, the HTML parser tool displays in pane 260 all hyperlinks of the site indicated by the web address. Significantly, however, neither this passage nor any other passages of ABRAMS describe or

suggest that any mapping is inspected in the process of displaying selected hyperlinks in a pane of the GUI of the HTML parser tool. In fact, in col. 4, line 21 ABRAMS expressly states that the program (i.e., the HTML parser tool) displays the selected hyperlinks, which suggests that any processing based on the user-specified URL (including displaying in the GUI) is performed internally within the executable code of the HTML parser tool without the need to reference any mapping such as the first mapping of Claim 18.

In contrast, Claim 18 includes the feature of determining, based on the first mapping, that an event output parameter associated with the particular event is mapped to a page parameter. When considered along with the other features of Claim 18, this feature indicates that the determination of an action is made based on a mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters. Since ABRAMS does not describe a mapping such as the first mapping of Claim 18 and since the ABRAMS' HTML parser tool does not even need a functionality that is based on such mapping, ABRAMS does not describe or suggest the above feature of Claim 18.

With respect to the above feature of Claim 18, on page 11 the final Office Action asserts that URL data representing an event is mapped to panes 240, 250, and 260 in Fig. 2A of ABRAMS, and that a display action with regards to the URL data is performed. This assertion is incorrect. As discussed above, ABRAMS expressly describes that the URL address is entered by a user into the GUI generated by the HTML parser tool. Thus, this URL address cannot possibly correspond to an event that is generated by a portlet as featured in Claim 18. Further, as also discussed above, the action of displaying content in the GUI of Figs 2A and 2B of ABRAMS is performed internally within the executable code of the HTML parser tool, and therefore such action is not determined based on a mapping such as the first mapping of Claim 18.

For the foregoing reasons, it is respectfully submitted that ABRAMS does not describe or

suggest the feature of Claim 18 of determining, based on the first mapping, that an event output parameter associated with the particular event is mapped to a page parameter.

- (6). ABRAMS does not describe or suggest the feature of Claim 18 of causing the action to be performed, wherein causing the action to be performed comprises passing a value of the event output parameter as the value of the page parameter.

Among other features, Claim 18 comprises:

causing the action to be performed, wherein causing the action to be performed comprises passing a value of the event output parameter as the value of the page parameter.

The final Office Action asserts that this feature of Claim 18 is described in col. 4, lines 21-29 of ABRAMS. This assertion is incorrect.

As discussed above, in col. 4, lines 21-30 and with respect to its Figs. 2A and 2B, ABRAMS describes that when a user selects a web address in pane 220 of the GUI, the HTML parser tool displays in pane 260 all hyperlinks of the site indicated by the web address. Significantly, however, neither this passage nor any other passages of ABRAMS describe or suggest that any value of an event output parameter is passed as the value of a page parameter in the process of displaying selected hyperlinks in a pane of the GUI of the HTML parser tool. In fact, in col. 4, line 21 ABRAMS expressly states that the program (i.e., the HTML parser tool) displays the selected hyperlinks, which suggests that any processing based on the user-specified URL (including displaying in the GUI) is performed internally within the executable code of the HTML parser tool. Further, this clearly indicates that ABRAMS' HTML parser tool does not even need any functionality that is performed in response to an external event and that is configured to resolve the external event by referencing a mapping such as the first mapping of Claim 18.

In contrast, Claim 18 includes the feature of causing the action to be performed, wherein causing the action to be performed comprises passing a value of the event output parameter as the value of the page parameter. When considered along with the other features of Claim 18, this

feature indicates that the action is determined made based on intercepted data that represents a portlet-generated event and on a mapping that maps one or more events to one or more actions and one or more event output parameters to one or more page parameters. Since ABRAMS does not describe a mapping such as the first mapping of Claim 18 and since the ABRAMS' HTML parser tool does not even need a functionality that responds to external events based on such mapping, ABRAMS does not describe the above features of Claim 18.

Finally, with respect to the assertion on page 11 in the final Office Action that URL data representing an event is mapped to the GUI panes in Fig. 2A of ABRAMS and that a display action with regards to the URL data is performed, it is respectfully submitted that the URL address (entered by a user in the GUI of Fig. 2A of ABRAMS) cannot possibly correspond to an event that is generated by a portlet (as featured in Claim 18) and that the display action to display contents (in the GUI of Fig. 2A of ABRAMS) cannot possibly correspond to an action that is determined based on a mapping in response to an event generated by a portlet (as featured in Claim 18).

For the foregoing reasons, ABRAMS does not describe or suggest the feature of Claim 18 of causing the action to be performed, wherein causing the action to be performed comprises passing a value of the event output parameter as the value of the page parameter.

For the reasons provided above in sections (1) through (6), ABRAMS and SCHAECK when taken alone or in combination do not describe or suggest all features of Claim 18. Thus, Claim 18 is patentable under 35 U.S.C. § 103(a) over ABRAMS in view of SCHAECK. Reconsideration and withdrawal of the rejection of Claim 18 is respectfully requested.

C. INDEPENDENT CLAIM 49

Claim 49 was rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over

ABRAMS in view of SCHAECK.

Claim 49 includes features similar to the features of Claim 1 discussed above, except in the context of a computer-readable medium. For this reason, it is respectfully submitted that Claim 49 is patentable under 35 U.S.C. § 103(a) over ABRAMS in view of SCHAECK for at least the reasons given above with respect to Claim 1. Reconsideration and withdrawal of the rejection of Claim 49 is respectfully requested.

D. INDEPENDENT CLAIM 66

Claim 66 was rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over ABRAMS in view of SCHAECK.

Claim 66 includes features similar to the features of Claim 18 discussed above, except in the context of a computer-readable medium. For this reason, it is respectfully submitted that Claim 66 is patentable under 35 U.S.C. § 103(a) over ABRAMS in view of SCHAECK for at least the reasons given above with respect to Claim 18. Reconsideration and withdrawal of the rejection of Claim 66 is respectfully requested.

E. DEPENDENT CLAIMS 2-17, 19-23, 47, 50-65, AND 67-72

Claims 2-3, 5, 7-8, 13-14, 16-17, 19-22, 50-51, 53-62, 64-65, and 67-72 were rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over ABRAMS in view of SCHAECK. Claims 4 and 52 were rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over ABRAMS in view of SCHAECK, and further in view of Hind et al., U.S. Patent Application Publication No. US 2004/0205555 (“HIND”). Claims 6, 9-12, 23, and 47 were rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over ABRAMS in view of SCHAECK, and further in view of Polizzi et al., U.S. Patent Application Publication No. US 2002/0052954 (“POLIZZI”). Claims 15 and 63 were rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over ABRAMS in view of SCHAECK, and further in view of Katariya et al., U.S. Patent No. 6,564,251 (“KATARIYA”).

Each of Claims 2-17, 19-23, 47, 50-65, and 67-72 depends directly or indirectly from one of independent Claims 1, 18, 49, and 66, and thus includes each and every feature of the independent base claim. Furthermore, in rejecting Claims 4, 6, 9-12, 15, 23, 47, 52, and 63 the Office Action relies explicitly on ABRAMS and SCHAECK, and not on HIND, POLIZZI, or KATARIYA, to show the features discussed above with respect to Claims 1, 18, 49, and 66. Because ABRAMS and SCHAECK do not teach the subject matter of Claims 1, 18, 49, and 66, any combination of ABRAMS and SCHAECK with the other three references necessarily fails to teach the complete combination recited in any dependent claim of Claims 1, 18, 49, or 66. Thus, each of Claims 2-17, 19-23, 47, 50-65, and 67-72 is allowable for the reasons given above for Claims 1, 18, 49, and 66.

In addition, each of Claims 2-17, 19-23, 47, 50-65, and 67-72 introduces one or more additional features that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case a separate discussion of those features is not included at this time. Therefore, it is respectfully submitted that Claims 2-17, 19-23, 47, 50-65, and 67-72 are allowable for the reasons given above with respect to Claims 1, 18, 49, and 66. Reconsideration and withdrawal of the rejections of Claims 2-17, 19-23, 47, 50-65, and 67-72 is respectfully requested.

II. CONCLUSION

The Applicants believe that all issues raised in the final Office Action have been addressed. Further, for the reasons set forth above, the Applicants respectfully submit that allowance of the pending claims is appropriate. Reconsideration of the present application is respectfully requested in light of the remarks herein.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

A petition for extension of time, to the extent necessary to make this reply timely filed, is hereby made. If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to charge any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,
HICKMAN PALERMO TRUONG & BECKER LLP

Dated: July 26, 2010

/StoychoDDraganoff#56181/
Stoycho D. Draganoff
Reg. No. 56,181

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Telephone No.: (408) 414-1080 ext. 208
Facsimile No.: (408) 414-1076